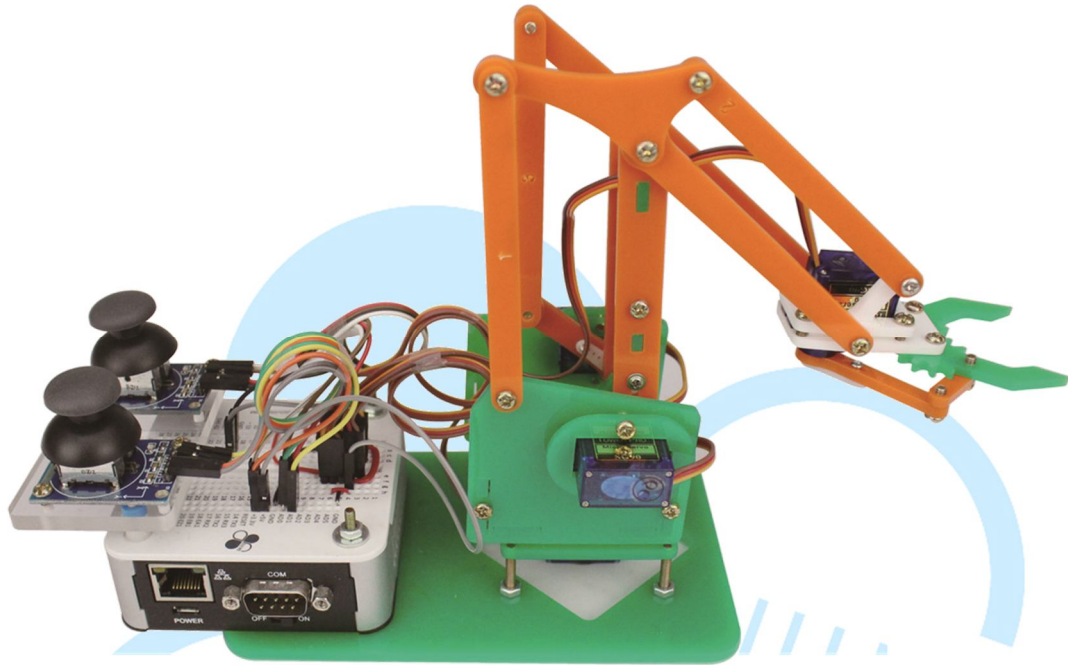


EduCake 控制機械手臂



一、關於機械手臂

機械手臂(RobotArm) 如圖 1 被廣泛的使用在各種工業生產環境中，主要是取代人手，去作繁重的工作(EX:汽車組裝、貨品裝箱)、重複性高的持續工作(EX:水果揀選)、精密要求的工作(EX:電路焊接)等等，這些工作人們並不是不能作，但機械手臂作來可以 24H 不用休息、不需要給薪水、不會耍脾氣而且還很精準不易出錯。所以在工業上已經有越來越多的公司採用機械手臂來取代人工的動作，只留下一些複雜的難以用機械手臂取代的事情才需要人工，可以大大的節省成本和增加產能、增加產品良率，這已經是產業的大勢所趨，所以之前

鴻海的郭董才會喊出要建立百萬機器人大軍。我們這章節就來談談機械手臂的



構成和控制。

圖 1. <http://www.kuka-robotics.com/> 很有名的 Kuka 機器手

二、 機械手臂的基本構造

工業用機械手臂動輒數十~數百萬台幣起跳，而且構造其實很複雜，主要可以分成機體本身、前端功能(可能是噴槍、焊槍、夾爪、吸盤、電磁鐵等等不同功能的東西，甚至需要能人工或是自動更換不同功能)、電源供應系統、PLC 控制系統、感測系統、電腦監視/警報系統、UI 控制介面等等部分所構成，這些東西通通要弄懂得要花費很久的時間。我們這裡先從最基本的教學用機械手臂開始來談。

員工會使用工業手臂之前得先教學作基本訓練，學校用在教學領域的機械手臂其實是很久以前就有人模仿工業手臂的長相，簡化很多之後作出各式各樣的教學用手臂，如圖 2，就是一種筆者很常用的機械手臂，使用一些很基本的機構和很容易買的到的馬達、控制板所構成，整個成本約是數千~一兩萬台幣就能買到。

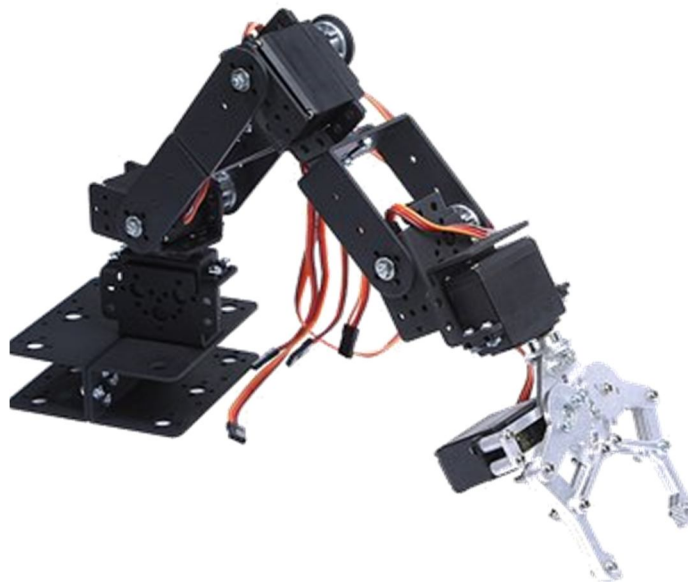


圖 2. 一種很常見的教學用機械手臂

這種機械手臂大約 10 幾年前就被創作出來，就是只有基本的功能，很多軸數控制產生各種動作，中間又一直有持續的改進，產生無數種衍生型別，筆者玩最多的就是這種類型，軸數約是 2~10 軸都有，也自己設計過十餘種不同的機械手臂。多年來基本的教學款式一直是類似這種的，直到 2013 年才開始有重大的演進。首先是因為網路募資的風行，使得一些想創業的 DIY 玩家看到機會，2013 年中左右，出現一隻叫作 uArm 的機械手臂，並且 2014 年初在 kickstarter 上面募資獲得空前成功，募得七百多萬新台幣

<https://www.kickstarter.com/projects/ufactory/uarm-put-a-miniature-in-dustrial-robot-arm-on-your>。因為這隻手臂的成功和他的開源設定(設計圖、電路、程式都完整分享到網路上)，募資結束後網路出現很多種模仿改進版本，像是 playArm、liteArm、meArm、xxArm...等等，其中的 meArm 就是我們本篇文章要採用的對象，作者把相關的資料都放在這裡公開給人取用

<http://www.thingiverse.com/thing:360108>，他的手臂長像約是如圖 3，被各方玩家用各種不同材質、顏色來製作呈現出來不同風格的作品。

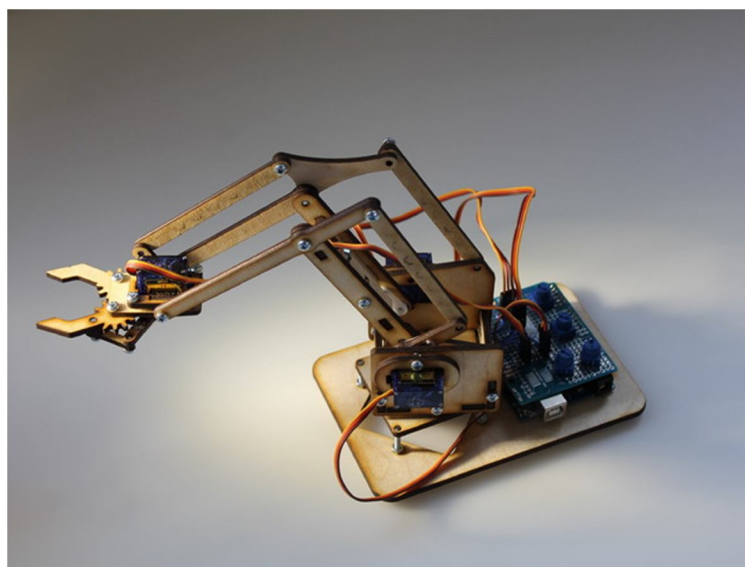


圖 3. 2014 年最受歡迎的教學機械手臂，這是木頭版本

為何說這隻手臂最受歡迎呢!? 因為他的構造非常簡單(明顯比 uArm 精簡，組裝好只要半小時)、能簡易的表達出工業機械手臂的運作原理、造價非常便宜、所有相關資訊都被作者公布在網路上。相對於三五年前還要價一隻一兩萬台幣的機械手臂來說，這隻只要兩千台幣出頭就能獲得的機械手臂迅速的獲得普通玩家、教學單位的注意和歡迎，而且他衍生出來的課程內容更是多不勝數。我們就用這隻手臂來談接下來的幾個主題吧。



三、 機械手臂的構造

meArm 類型的機械手臂主要的構成有機構、馬達、電路板、供電系統、控制程式等等。機構在網路上已經有人分享相關的各種版本的雷射切割檔，可以輕易使用 3mm 厚度的壓克力、鋁合金、木頭等等材料切出來組裝，他也有 2~5mm 的版本，如圖 4. 是使用了各種顏色的 3mm 壓克力雷射切割出來的手臂零件。

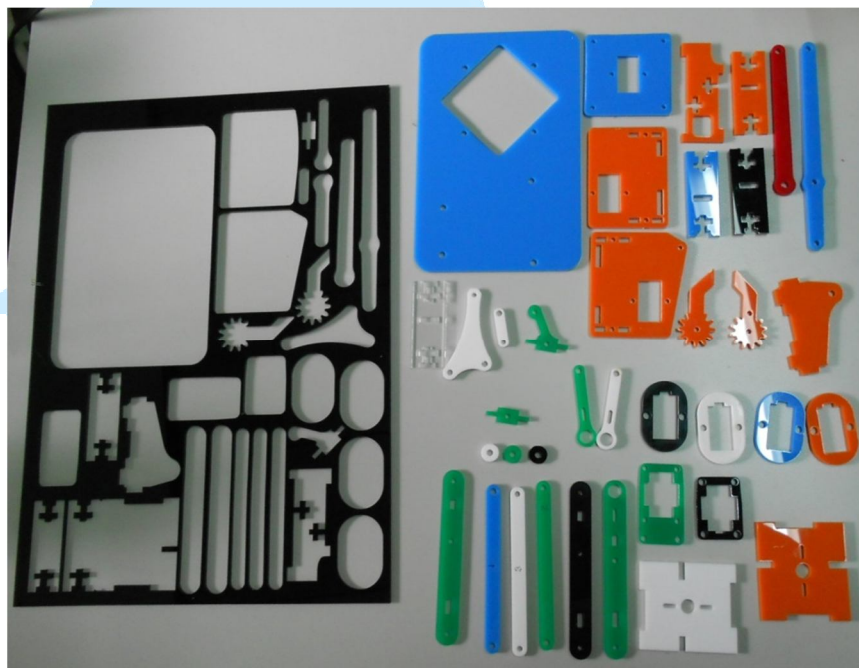


圖 4. 所有需要的零件，使用壓克力切割

這裡使用的材料厚度一定要 3mm 的板材，不然就要改設計，主要是原始設計的圖檔有不少地方使用類似木頭隼合加螺絲螺母鎖住的方式來設計製作，若厚度稍小，組裝起來會搖搖晃晃；厚度稍大，則根本就組裝不起來了，可能得借助挫刀之類的工具來擴孔才有辦法，算是這種類型手臂的重大設計瑕疵，畢竟板材是有工差的，有些板材公差稍大就會有這問題。

馬達主要是使用 SG90，這顆馬達在前面的 PWM 章節裡面已經有介紹過規格和控制方式，採用他主要是因為便宜和容易取得。但筆者對這顆馬達其實是很多意見的，功能的部分先不說，主要是馬達外殼的公差實在太大了，買了一千顆馬達進來，外殼的大小大概可以差距到 1mm，這對機構的設計會造成很大影響，孔位抓太小，萬一遇到較大的馬達外殼會裝不下去；孔位設計大一點，遇到較小的馬達外殼，裝下去會很鬆垮的感覺，且還會造成馬達轉軸位置的誤差，導致手臂運作不穩。國外的原設計作者也有發現這個問題，他的解法就是用另外一片機構去用夾的，勉強能處理掉這個，但組裝時就是要小心調整就是了。本章節主要是介紹機械手臂入門，重點在快速上手使用，先用這顆，後面我們要談精密控制時，會重新使用類似 SIZE 的更高階版本馬達來重新設計機構，這對於手臂運作穩定度和精密度是有很重大幫助的，預計會使用這顆馬達來實作 http://www.roboard.com/servo_0263.html，速度更快、力量更大、精密度也明顯高，還是金屬作的。而且這時候 EduCake 的 13bit 的 PWM 高精度就能發揮用途，讓控制更順暢精密，這也是使用 SG90 這種初級馬達作不到的事情。

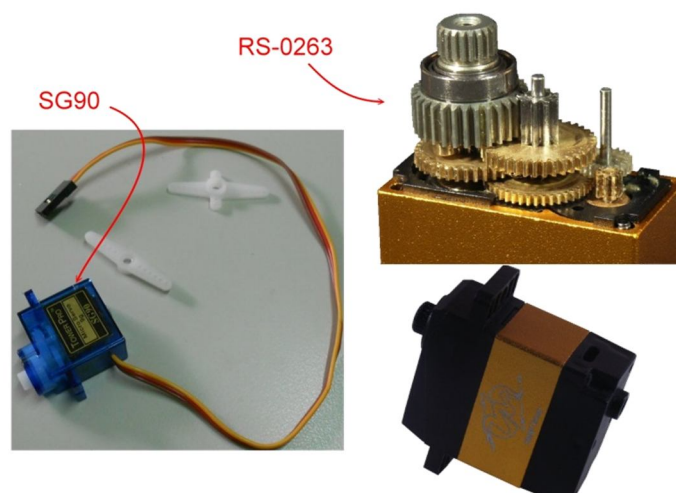


圖 5. SG90 和精密穩定許多的 RS-0263

控制電路板主體當然是使用我們的主角- 86duino EduCake 啦，但是為了方便使用者直接操作，也需要使用搖桿等等的周邊模組，如圖 6.

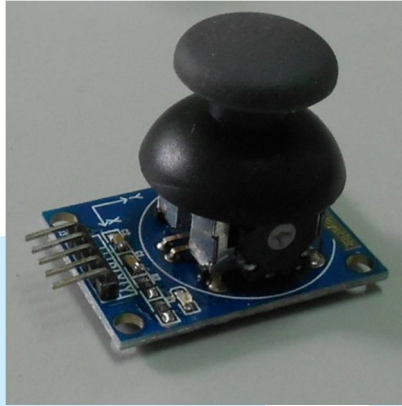


圖 6. 可以控制前後左右，和一個按鈕功能的搖桿模組

這種模組很容易使用，他主要是有 5 個腳位，分別需要接到: Vcc(5V)、GND、X、Y、button。其中的 X、Y 就是接到 analog 腳位，以類比的方式來讀取搖桿前後左右的數值，其實他內部構造就是 10K 歐姆的 VR 而已，一般來講是介於 0~1023 的傳回值，或是使用 EduCake 特有的 `analogReadResolution()` 指令來獲得 11bit 解析度或是 0~2047 的數值。

但這種模組通常製作的零件有誤差或是他裝的有點歪曲，正中點並不是預想的 512 這個數值，而可能是 470 或是 600 之類的離譜數字，所以使用前要先測試看看都不碰他的時候傳回的數值到底是多少來決定中點在哪裡，且測試出中點假設在 500，需要設定一個不動作區間，EX:470 ~ 530 這個範圍都不動作，這樣可以避免輕輕觸碰手臂就亂動，這對手臂一開始的初始化很重要。當然後面我們也可以寫更複雜的演算法來處理這種事，至於 button 就是一顆小型的按鈕藏在搖桿模組裡面，不過這顆按鈕很難按，有點硬，而且他是直接裝在搖桿正下方，按下的時候難免會影響 XY 的數值稍微的偏離，連帶手臂會搖

晃，一般我們有需要時不太會採用這顆按鈕，而是另外使用單獨的按鈕來作想要的功能。

最後，把這一切組裝起來就變成圖 7 這樣，組裝過程請自行參照前面講過的網路上的文章分享，裡面就有詳細步驟，且所附的 arduino 原始碼也可以直接拿來 EduCake 使用沒有問題(腳位要稍微確認一下是不是一樣)，當然這裡我們自己寫的效果更好，還可以針對性的發揮不同的功能。

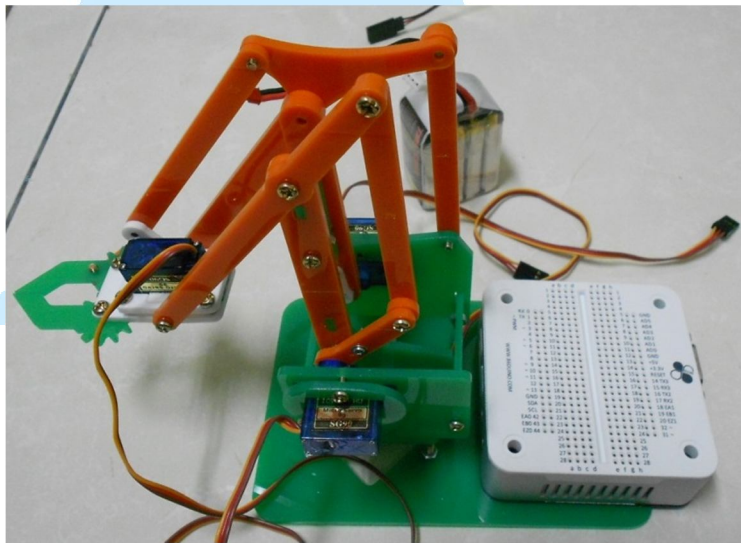


圖 7 機械手臂組裝好了以後的長相

組裝好的手臂，後臂棕色部分的馬達旋轉，會帶動整個手臂前半部包括夾爪運動，如圖 8，且運動過程中，這種平行機構的特點是，不管後臂如何運動，手臂前半部含夾爪，也就是圖中綠色的部分，他的形狀是不會有任何改變的，對於寫程式作逆運動學計算會很方便。

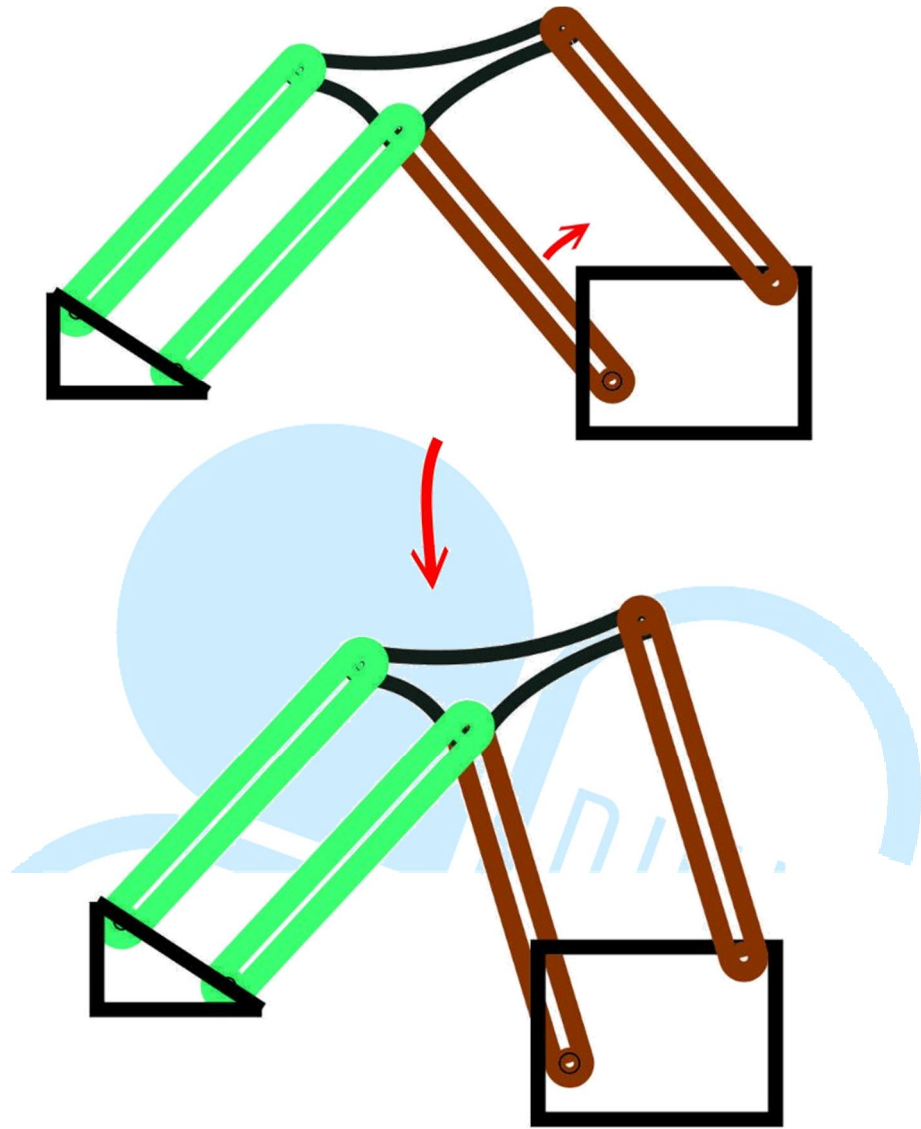


圖 8 後臂運動的狀況

圖 9 是前臂的運動狀態圖，由圖中的紅色部分的馬達來帶動，主要是可以控制夾爪的上下。其實可以不用那麼麻煩用連桿，直接把馬達裝到關節處也行(圖中紅點處)，但因為這種設計主要是用在工業用途，通常為了出力要大，馬達會很大顆且很重，若是裝到關節的地方，會造成手臂水平伸直的時候，給後臂的馬達造成太大的力矩負擔，導致必須更換更大的馬達且損耗也會變快，使用連桿就不會有這個問題。

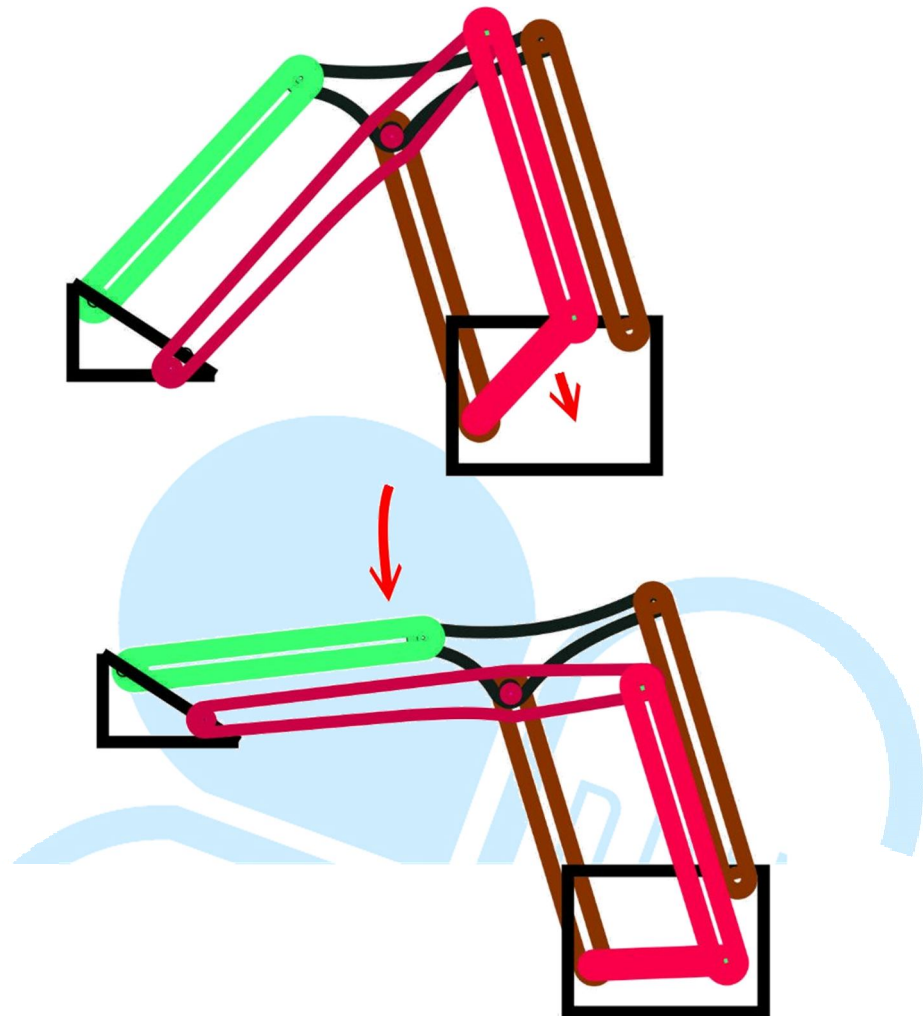


圖 9 前臂運動的狀況

圖 10 是夾爪部分的馬達，原設計使用連桿來帶動夾爪的夾取，這種設計相對於之前圖 2 的機械手臂所使用的平行夾爪，好處主要只有體積較小和夾取範圍稍大。但他使用連桿機構來帶動夾爪，有幾個重大的設計瑕疵，第一個就是因為螺絲孔位的問題，固定好以後的夾爪依舊會有點明顯搖晃，再來就是馬達帶動的那個連桿過長，使得馬達的角度改變一點點，夾爪就會夾住或放開，對於精密的夾取動作比較作不出來，實際測試約是馬達改變 20 度角，夾爪就會由張開最大的程度變成完全閉合的程度，平行夾爪在同樣的情況下馬達約要改變 100 多度左右才能完全開合，精密度相較之下有好幾倍的差距，所以 meArm 這個設計主要只能

對物品做夾或是放開的動作，使用上就要小心不要夾太緊、夾太久，會讓馬達因為長時間通過大電流而燒毀(馬達本身無保護機制)。

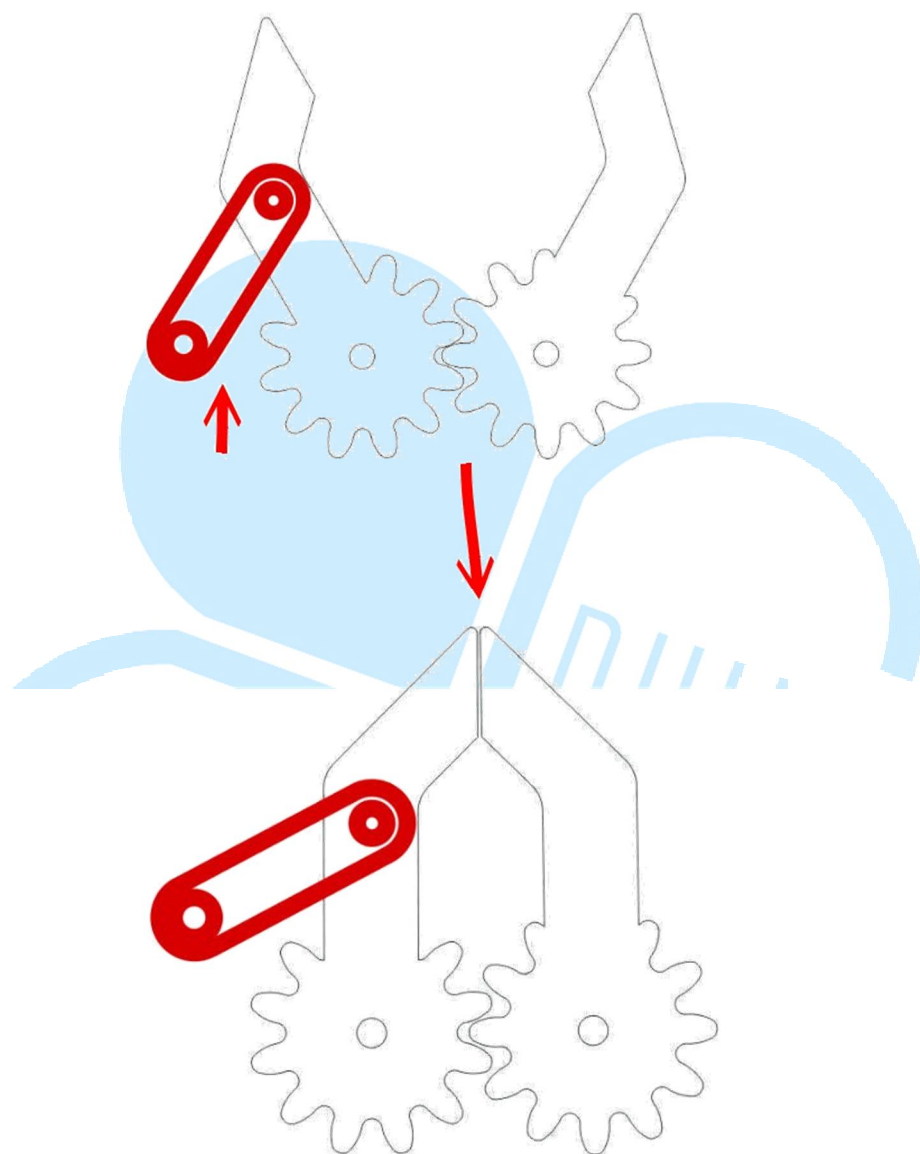


圖 10 夾爪的運動狀況

以上幾張圖可以簡易的了解整個手臂所有節點的基本運動狀況，這部分筆者比較喜歡直接用冰棒棍量取一樣的長度+大頭針先作出基本形狀來測試每一軸的運動狀況以及可能對旁邊的機構或是整個手臂造成的影響，來慢慢修正架構，效果感覺還不錯。

四、 機械手臂的基本控制

前面介紹大致的架構，並組裝出一隻基本的四軸機械手臂，其中最重要的一部分就是組裝過程的馬達角度定位。因為 SERVO 只能在中點往左往右運動數十度的範圍，SG90 約正負 80 度範圍內比較安全，且也得考量機構之間的干涉問題(運動到卡住或是干擾到別的零件的運作都是)，所以在每一軸要安裝的時候，得要先在紙上或是使用繪圖軟體先模擬這一軸想要的運動範圍，然後把馬達的位置調整到中點才安裝，如圖 11，圖中的步驟 1，就是先要確認這一軸的運動範圍和他的中點在哪個位置；步驟 2，先使用簡單的馬達測試電路(請參考 PWM 章節)，給這顆馬達 1500us 的位置讓他置中；步驟 3，把該機構安裝到馬達上面，安裝過程馬達的擺臂必須鎖緊馬達不可取下，避免定位好的中點又跑掉。

這動作對初學者來講比較困難，因為中點位置可能是偏移水平方向或是垂直方向一點點，不容易準確抓到，所以比較好的機械手臂機構設計是必須把這考慮進去，讓機構在馬達置中的情況下，可以簡易的直接使用垂直或是水平方向安裝，對定位。

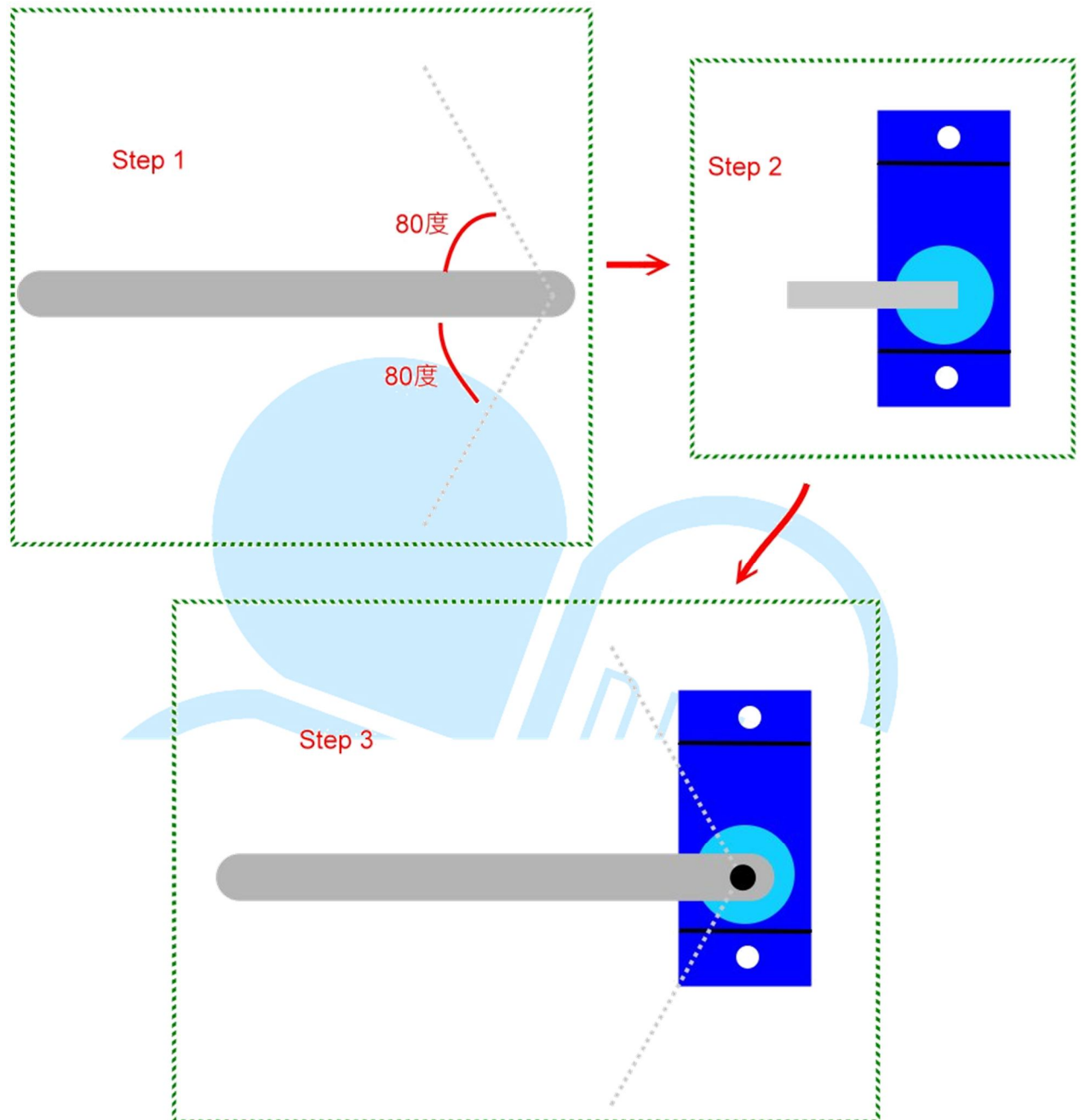


圖 11. 每一軸馬達的安裝方式

至於讓馬達到中點的程式碼就很簡單了，如下的程式就可以，或是買現成的馬達調整電路也行。

```
#include <Servo.h>
Servo myservo;
void setup()
{
    myservo.attach(3); // 連接 digital 腳位 3
}
void loop()
{
    myservo.write(90); //移動到 90 度的位置
    // 或使用 writeMicroseconds(1500); 移動到 1500us 是更好
    的選擇
}
```

一般除了作這個置中定位的動作外，馬達和機構鎖上後，還會額外讓馬達左右慢慢擺動到極限去測試看看在不同的角度下馬達是否有干涉到別的機構或是突出的螺絲(這在 solidwork 等繪圖軟體裡面也可以模擬)。也可以使用額外的一顆 VR 加裝到 EduCake 去讀取 VR 的數值來讓馬達轉動到想要的位置，順便輸出到監控視窗來方便記錄馬達運動的左右極限值，後面寫程式的時後會用到，簡易的測試程式碼如下：

```
#include <Servo.h>
Servo myservo;
void setup()
{
    myservo.attach(3); // 連接 digital 腳位 3
    Serial.begin(9600);
}
void loop()
{
    int a;
    a=analogRead(0); // VR 的腳位裝在 analog 腳位 0
    a=map(a,0,1023,0,180); // 把讀取到的 VR 值對應到 0~180
    度
    // 這裡要注意的是，開機前必需確認 VR 的位置在差不多中點
    處
    // 若 VR 的位置在兩端，可能開機後馬達會立刻撞到卡住
    // 會損壞馬達
    myservo.write(a); //移動到 90 度的位置
    // 或使用 writeMicroseconds(1500); 移動到 1500us 是更好
    的選擇
    Serial.println(a); // 順便顯示到監視視窗，以便記錄下目前的
    位置}
```

組裝的定位好了以後，把電路也接好，手臂會如圖 12，其中一顆馬達負責底座旋轉，圖中編號 1 的馬達負責後臂的前後運動；編號 2 的馬達負責前臂的前後運動；編號 3 的馬達負責夾取物品。

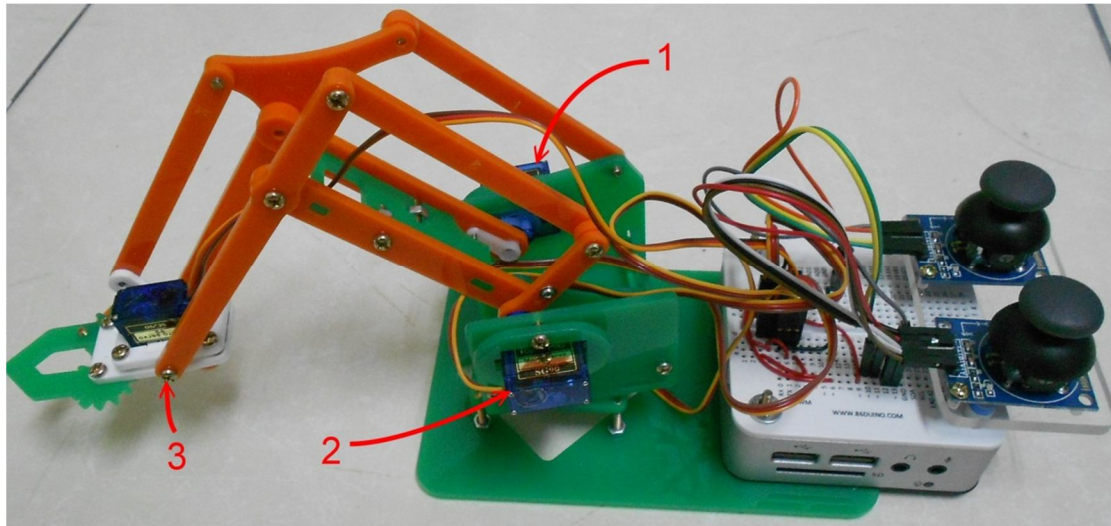


圖 12. 手臂結合電路以後的樣子

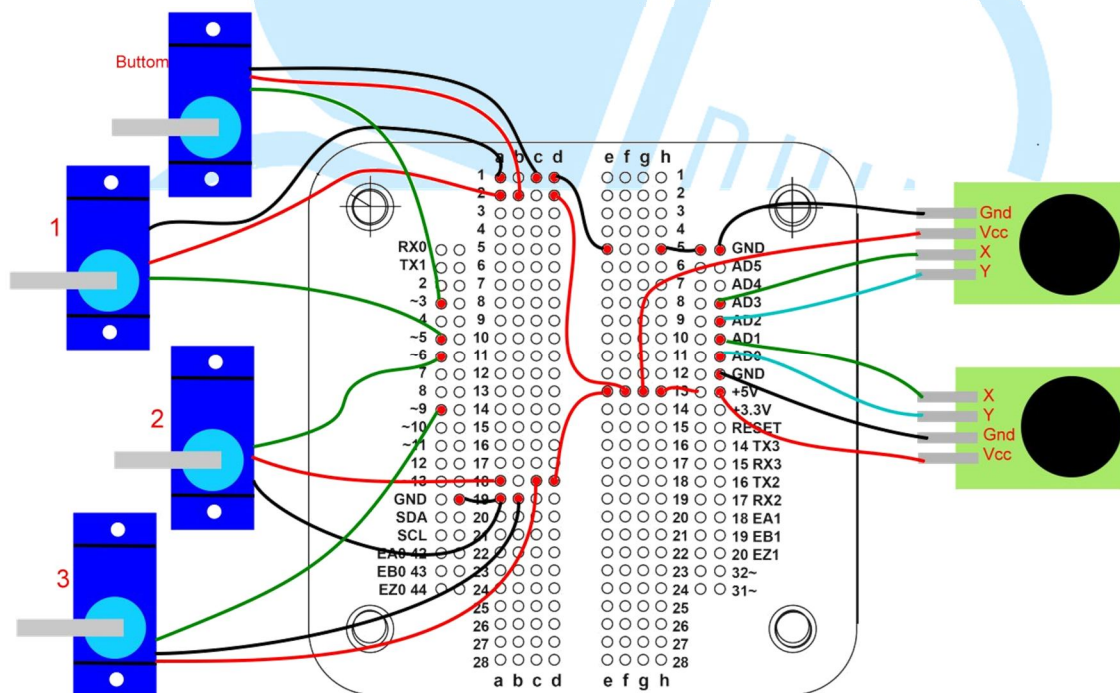


圖 13. 接線圖

圖 13 是接線圖，馬達使用 digital 腳位的 3、5、6、9 四個 PWM 輸出腳位，兩組搖桿的 XY 接到 analog 腳位 0~3，分別對應控制手臂的左右旋轉、前後臂的運動和夾爪等四個功能。

五、 機械手臂的整合控制

整隻手臂完成組裝和馬達定位以後，就可以開始來整合整個程式

```
#include <Servo.h>

Servo myservo1; // 底盤旋轉馬達
Servo myservo2; // 後臂升降馬達
Servo myservo3; // 前臂馬達
Servo myservo4; // 夾爪馬達

void setup()
{
  Serial.begin(9600);
  myservo1.attach(3);
  myservo2.attach(5);
  myservo3.attach(6);
  myservo4.attach(9);
}

void loop()
{
  int a,b,c,d,e;
  // 由 abcd 四個變數先讀取搖桿的數值
  // 並依照之前講的測試方式取得的馬達運動角來控制對應的馬達
  // 因為每個搖桿模組的狀況並不相同，以下數值僅供參考
  a=analogRead(0); // 搖桿 1 由左至右對應 1023~ 0, 中點約在 497
  b=analogRead(1); // 搖桿 1 由後往前對應 0~1023, 中點約在 508
  c=analogRead(2); // 搖桿 2 由後往前對應 0~1023, 中點約在 490
```

```
d=analogRead(3);// 搖桿 2 由左至右對應 1023~0，中點約
500
Serial.print(a); // 輸出到監控視窗方便查看
Serial.print(",");
Serial.print(b);
Serial.print(",");
Serial.print(c);
Serial.print(",");
Serial.println(d);

// 底座控制，因為是置中安裝上去，運動角度對應到 20~160
度
a=map(1023-a,0,1023,20,160);
// 直接把轉換過的角度寫入 servo1，就可以控制底座旋轉
myservo1.write(a);
Serial.print(a);
Serial.print(",");

// 後臂控制，運動角度對應到 90~150 度
b=map(1023-b,0,1023,95,150);
Serial.print(b);
Serial.print(",");
myservo2.write(b);

// 前臂控制，運動角度對應到 40~140 度
c=map(1023-c,0,1023,40,140);
Serial.print(c);
Serial.print(",");
myservo3.write(c);

// 夾爪控制，打開和閉合的運動角度對應到 70~90
d=map(1023-d,0,1023,70,90);
Serial.print(d);
```

```
myservo4.write(d);  
  
delay(15);  
}
```

底座的旋轉就是中央往兩邊各約 70 度左右的旋轉角。前後臂也是置中安裝上去，但因為機構間彼此的限制，旋轉角度就會比較特殊；夾爪的部分就更小了，只有 70~90 度角之間。這些數字會因為換馬達或是換了不同的搖桿模組，會有不一樣的結果，需要重新測試修改。另外上面主要使用 **myservo.write()**；這個指令，他裡面的參數是使用角度值，但這個角度值其實也只是參考，不同種類的馬達是不一樣的，他並不是真的“度”這個單位，EX: 使用 SG90 給 120 度，可能是從中央往左轉 30 度(90 度 - - > 120 度)，但同樣的指令給 MG995 這種馬達，可能只轉了 26 度，有些馬達甚至會轉到 60 度這麼多，所以才說不管使用哪一種馬達或是模組都得要重新測試才行。若是覺得這個 **myservo.write()**；指令不夠精確，那就換用 **myservo.writeMicroseconds ()**；改用脈波來控制，但這需要馬達精確度要能跟上，SG90 不需要使用這個指令，因為他本來就不太準。

這兩個指令一個是角度控制，一個是脈波控制，其實內部實作都是轉換成脈波來控制馬達，兩者的對應最大範圍是 0~180 度，約是對應到 500~2500us(或是 700 ~2300us)，這兩組對應一樣只是參考，有些馬達是 20~160 度對應到 800~2200us，有些是 35~145 度對應到 1000~2000us，有些則是 25~155 度對應到 900~2100us，差別真的很大。通常馬達在使用的時候一定要先詳閱說明和限制，有些馬達超過指定的範圍是會卡住甚至快速燒毀的，都必須注意。

六、 結語

機械手臂是一種很好玩的東西，拿來玩、教學或是工業/生活的應用都是很好的方向，這篇文章就當作我們一系列機械手臂教學文章的引子，後面我們會來談更多的進階應用，像是自動裝箱、自動排列、自動搬運等等更有趣和實用的主題。

